

DIMM-SH7723-ADK-CE 6.0 R3

User manual

Copyright



© Copyright 2010 emtrion GmbH

All rights reserved. Without written permission this documentation may neither be photocopied nor stored on electronic media. The information contained in this documentation is subject to change without prior notification. We do not assume any liability for erroneous information or its consequences. The trademarks of other companies that are used identify the products of these companies exclusively. Microsoft, Windows, Windows95, Windows98, Windows NT, Windows 2000, Windows XP, Windows CE and MS-DOS are registered trademarks of the Microsoft Corporation.

Revision No.	Changes	Date
1	First edition	17.03.2009/MI
2	<ul style="list-style-type: none"> - Improving stabilisation and performance of NAND - Synchronising of AUDIO - Replacing of USBH driver with DMA support - Supporting additional display UMSH8272 - Reset of the hardware can be triggered from software - Graphical hardware accelerator - Multimedia <ul style="list-style-type: none"> Video IN(PAL, NTSC, digital) Video OUT(PAL, NTSC) Video decoding MPEG4 Blending PIP - QFEs followed up to 2009M05 - New section: Licensing of the WINCE-Image - IIC-Bus driver: "Repeated start condition" supported - Sample applications for multimedia 	30.06.2009/MI
3	<ul style="list-style-type: none"> - Licence agreements replaced - QFEs followed up to 2009M08 - SDCard: <ul style="list-style-type: none"> Both interfaces, SDHI0 and SDHI1 of the SH7723 are supported. High Capacity is now supported - New multimedia demo application. Available as msi-file. - Update video decoder → new revision supported - Some small improvements 	11.09.2009/MI
4	<ul style="list-style-type: none"> - Using Windows CE 6.0 R3 - QFEs followed up to 2010M03 - Updated USB Host driver - Support of Hive-based registry 	07.06.2010/Ft

This document is published by:

Emtrion GmbH

Greschbachstr. 12

D-76229 Karlsruhe

Tel: +49 (721) 62725-0

Fax: +49 (721) 62725-19

E-mail: mail@emtrion.de

Internet: <http://www.emtrion.de>

June2010-004

Index

1	Introduction	7
2	DVD of the ADK	8
3	Requirements	9
3.1	Requirements on the host	9
3.1.1.	SW-Requirements	9
3.1.2.	HW-Requirements	9
3.2.	Requirements on the target	9
4.	Setting up the host	10
4.1.	Installing the SDK	10
4.2.	Setting up Ethernet(TCP/IP)	10
4.3.	Setting up a TFTP server	10
4.4.	Setting up a Terminal	10
5.	Setting up the target	12
5.1.	Setting up for application development	12
5.2.	Setting up for updating the image	12
6	BSP contents	13
6.1	Applications – End User	13
6.2	Applications and Services Development	13
6.3	Communication Services and Networking	14
6.4	Core OS Services	14
6.5	Device Management	15
6.6	File Systems and Data Store	15
6.7	Fonts	15
6.8	Graphics and Multimedia Technologies	15
6.9	International	16
6.10	Internet Client Services	16
6.11	Security	17
6.12	Shell and User Interface	17
6.13	Device Driver	17
7	Application development	19
7.1	VS2005 for Application Development	19
7.1.1	Creating a new Managed Project	19
7.1.2	Deploying to the target	19
8	Starting an application at system start	23
8.1	Emtrion's autostart mechanism	23
8.2	Autostart with launchXX and dependXX	24
9	Data Exchange between Target and Workstation	26
9.1	Preparing the Desktop Computer	26
9.2	Preparing the Windows CE Device	26
9.3	USB ActiveSync	26
9.4	Remarks on Using ActiveSync	27
10	Multimedia sample applications	31
10.1	UMP_600_V2.exe	31
10.2	Videointest.exe	31
11	Persistent Registry	32
11.1	Deleting the Persistent Registry	33
11.1.1	Deleting the Registry by the Bootloader	33
11.1.2	Deleting the Registry by means of an Application under Windows CE	33

12	Downloading the image to the target	34
12 .1	Perform downloading	34
13	The Bootloader	35
13 .1	Location of the bootloader	35
13 .2	Communication settings	35
13 .3	Console mode	35
13 .4	Boot mode	35
13 .5	Bootmenu	35
13 .6	Get to the bootloader prompt	36
13 .7	Changing the display	37
13 .8	Supported File Formats	37
14	Serial Ports	37
15	I²C Bus Driver	39
15 .1	General Information	39
15 .2	Implementation	40
15 .3	API	40
15.3.1	CreateFile	40
15.3.2	DeviceIoControl	41
15.3.2.1	IOCTL_I2CBUS_SETADDRESS	41
15.3.2.2	IOCTL_I2CBUS_READWRITE	42
15.3.3	ReadFile	43
15.3.4	WriteFile	43
15.3.5	CloseHandle	44
15.3.6	Error Codes	44
16	Licence Agreements	46

1 Introduction

This application package addresses all software engineers who don't want to care about the details of the Windows embedded CE 6.0 R3 kernel and who directly want to start writing applications. That's why emtrion GmbH bundles packages with all the files which are necessary for writing applications. Mainly a kernel and its associated SDK with a typical set of components are available. It is based on rich experiences with different customer specific kernels and so it is suitable for the most applications. The kernel is a GUI based kernel. The details of the kernel are depicted in latter chapters.

Anyway, if the kernel doesn't fit to the applications needs there are two possibilities to fix this problem. The first one is that the customer purchases a SYSDeveloper Kit and does all the modifications of the BSP by means of the MS Platform Builder by himself or he asks emtrion GmbH to do this work for him.

The following chapters of this documentation give a brief overview about all what is good to know when starting writing applications for emtrion's DIMM-SH7723. Parts of the following chapters are excerpts of the documents which are part of the Developer Kit packages.

Important:

The kernels coming with this package are not licensed and have a limited live time. You can use them for demonstration, tests and application development.

If you need a kernel for your series production it is mandatory to activate the kernel by emtrion GmbH and to have a license from Microsoft for each CPU on which this kernel is running. Please contact emtrion GmbH about the details of licensing and on how to get such licenses from Microsoft.

2 DVD of the ADK

The content of the DVD is tailored for application development and is listed in the table below.

Folder	Filename	remarks
manuals	- HiCODIMM7723_ADK_CE600_v...en.pdf - HiCODIMM-ECOBBase_v...en.pdf - HiCODIMM-Base_HW_v...en.pdf - manual_uboot_hico7723_v...en.pdf - UMP_User_Developer_Documentation_v1en.pdf - CEU_Driver_user_Documentation_v1en.pdf	This manual. Hardware manual of the HiCODIMM-EcoBase* Hardware manual of the HiCODIMM-Base* Manual of the bootloader Includes description of the multimedia application ump_600_v2.exe Includes description of the multimedia application videointest.exe
Kernel	wce-hico7723	Image of the licensed kernel.
Sdk	HiCODIMM7723DevKit_CE600_SDK	Software Development Kit for the kernel
Usbdriver	Wceusbsh.inf Wceusbsh.sys	inf- file for the Microsoft ActiveSync driver Microsoft ActiveSync driver

(*) Dependent on the delivered edition of the ADK

3 Requirements

3.1 Requirements on the host

The requirements you need for application development are mentioned below.

3.1.1. SW-Requirements

- Operating System
 - Windows® XP(SvP2)- or Windows® 2000(SvP4)
- Microsoft ActiveSync Version 4.2.0 for development purpose and data exchange between target and host
- Microsoft Visual Studio 2005
- When you want to update the image a TFTP-Server is required too. One can be downloaded for free for example at http://tftpd32.jounin.net/tftpd32_download.html. When you are using this tftp, be sure its dhcp service is disabled to avoid conflicts with the current dhcp at your network.

When you want to update the image a TFTP-Server is required too.

3.1.2. HW-Requirements

- Speed: 933 MHz, recommended 2 GHz
- RAM: 512 MB, recommended 1 GB
- Interfaces
 - 1x Ethernet port
 - 1x RS232
 - 1x USB 1.1/2.0
- DVD-ROM drive

3.2. Requirements on the target

For the target the following devices are recommended:

- USB keyboard
- USB mouse

4. Setting up the host

At this point we assume Microsoft® Visual Studio® 2005 with SvP1 Microsoft® and Microsoft® ActiveSync are already installed on the host.

4.1. Installing the SDK

The SDK allows you to develop applications based on the delivered kernel. To make this possible install the SDK from the DVD when VS2005 is not open. During installation, the SDK is added to the list “Installed SDKs” from VS2005. At this point HiCODIMM7723 resp. DIMM-SH7723 can be chosen in VS2005 for generating applications on DIMM-SH7723.

4.2. Setting up Ethernet(TCP/IP)

TCP/IP is used for Ethernet connection to DIMM-SH7723. Make sure that TCP/IP is already set for the network protocol at the host.

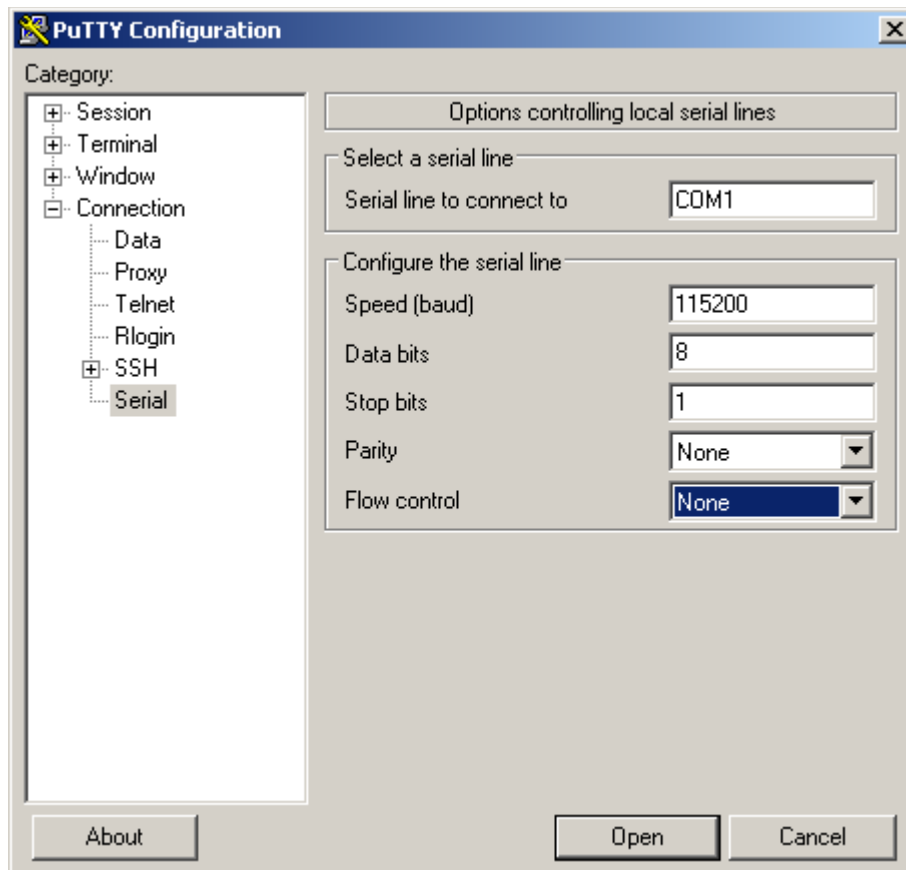
4.3. Setting up a TFTP server

This is only required when you intend to update the image on the target. In this case make sure the TFTP is running and its browse directory points to the location of the image. A link for a TFTP server is stated in the section [SW-Requirements](#).

4.4. Setting up a Terminal

As terminal software, use the terminal software that comes with your host or any software you are already using. In the terminal setting window, set the communication protocol as follows.

- (1) Use the supplied serial cable to connect serial port COM1 on the target to the host
- (2) Start the terminal software on the host and make the settings following below.
 - Baud rate 115200 bps
 - Data length 8 bits
 - Parity None
 - Stop bit 1
 - Flow control None



5. Setting up the target

This Developer Kit is delivered with the core module DIMM-SH7723 and dependent on the scope of the delivery with one of the two available HiCO.DIMM- base boards. To make a fast operation possible the target is already assembled. For developing purpose only a few steps are necessary.

Please note that everything is discharged when touching the target.

5.1. Setting up for application development

There are only two steps to do.

- Connect the target to the host via the supplied USB cable A-B to allow a Microsoft ActiveSync connection. The USB-B type connector is located at the front of the target.
- Use the Ethernet cable to connect the target to the network where the host is linked to.

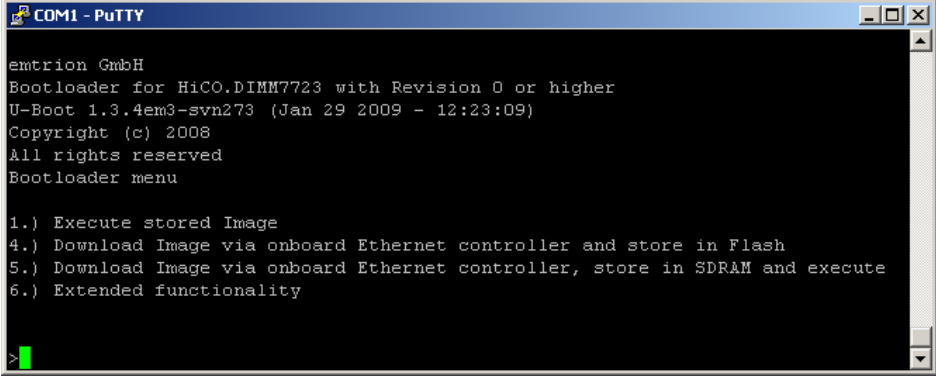
5.2. Setting up for updating the image

- Configure the bootloader at the target for using the console. For it, turn off the switch SW1 -4 on DIMM-SH7723

DIP Switch SW1:

	4	3	2	1
off	X	X	X	X
on				

- Connect the target to the host via the serial cable. Use the 9-pin D-Sub jack of serial port UART A at the target.
- Power ON or reset the target. When a terminal program is running on the host the bootloader outputs the following menu.



```

COM1 - PuTTY
emtrion GmbH
Bootloader for HiCO.DIMM7723 with Revision 0 or higher
U-Boot 1.3.4em3-svn273 (Jan 29 2009 - 12:23:09)
Copyright (c) 2008
All rights reserved
Bootloader menu

1.) Execute stored Image
4.) Download Image via onboard Ethernet controller and store in Flash
5.) Download Image via onboard Ethernet controller, store in SDRAM and execute
6.) Extended functionality

>
  
```

- Make a physical Ethernet connection, like in application development above. Set the network environment variables to your requirements, like "dhcp" or "ipaddr" and "netmask". Assign the IP-address of the TFTP server to "serverip". The settings have to be made at the level of the bootloader prompt. [Get to the bootloader prompt](#)

6 BSP contents

This chapter describes briefly the contents of the kernel delivered by CD or available online at our support site. The kernel is a typical one that should be suitable for most of the applications. If modifications are necessary, you may consider purchasing a SYS-Developer Kit from emtrion or just asking emtrion to design an individual kernel for your needs.

6.1 Applications – End User

Kernel		ADK
Active Sync		X
File Sync		X
CAB File Installer/Uninstaller		X

6.2 Applications and Services Development

Kernel		ADK
.NET Compact Framework 2.0		
.NET Compact Framework 2.0		-
.NET Compact Framework 2.0 – headless		-
OS Dependencies for .NET CF 2.0		-
OS Dependencies for .NET CF 2.0 - headless		-
.NET Compact Framework 3.5		
.NET Compact Framework 3.5		X
.NET Compact Framework 3.5 – headless		-
OS Dependencies for .NET CF 3.5		X
OS Dependencies for .NET CF 3.5 – headless		-
Active Template Library (ATL)		X
C Libraries and Runtimes		
C++ Runtime Support for Exception Handlind and Runtime Type Information		X
Full C Runtime		X
Standard I/O (STDIO)		X
Standard I/O ASCII (STDIOA)		X
Standard String Functions - ASCII		X
COM and DCOM		
DCOM, incl. COM Storage		X
Minimal COM (No OLE Support)		
String Safe Utility Functions		X
XML		
XML Core Services and Document Object Model		X
XML Query Languages (XQL)		X

6.3 Communication Services and Networking

Kernel		ADK
Networking General		
NDIS User-mode I/O Protocol Driver		X
Network Driver Architecture (NDIS)		X
Network Utilities (Ipconfig, ping, route, netstat)		X
TCP/IP		X
IP Helper API		X
TCI/IPv6 Support		X
Windows Networking API/Redirector (SMB/CIFS)		-
Winsock Support		X
Networking – Local Area Network (LAN)		
Wired Local Area Network (802.3, 802.5)		X
Networking – Wide Area Network (WAN)		
Dial Up Networking (RAS/PPP)		X
AutoDial		X
Standard Modem Support for Dial Up Networking		X
Telephony API (TAPI 2.0)		X
Unimodem Support		X
Servers		
Core Server Support		X
File Server (SMB/CIFS)		-
FTP Server		X
RAS Server/PPTP Server (Incoming)		-
SNTP Client with DST		X
Telnet Server		X

6.4 Core OS Services

Kernel		ADK
Toolhelp API		X
Device Manager		X
Display Support		X
Serial Port Support		X
UI Proxy for Kernel-Mode Drivers		X
Windows Embedded CE Driver Development Kit Support Library		X
Kernel Functionality		
Fiber API		X
Format Message API		X
Format Message API – System Error Messages		X
Memory Mapped Files		X
Message Queue – Point-to-Point		X
Target Control Support (Shell.exe)		X
Notification		
Non UI based Notification		-
UI based Notification		X
Power Management		
Power Management (Full)		X
Power Management (Minimal)		-
USB Host Support		
USB Function Driver		X
USB Host Support		X

USB Human Input Device (HID) Class Driver		X
USB HID Keyboard and Mouse		X
USB HID Keyboard only		X
USB HID Mouse only		X
USB Printer Class Driver (PCL-3)		X
USB Storage Class Driver		X

6.5 Device Management

Kernel		ADK
Simple Network Management Protocol (SNMP)		X
Device Management Client		X

6.6 File Systems and Data Store

Kernel		ADK
Compression		X
Database Support		
CEDB Database Engine		X
File and Database Replication, bit-based		X
File Cache Manager		-
File System – Internal		
RAM and ROM File System		X
ROM-only File System		-
Registry Storage		
Hive-based Registry		X
RAM-based Registry		-
Storage Manager		
FAT File System		X
Partition Driver		X
Release Directory File System		X
Storage Manager Control Applet		X
System Password		X

6.7 Fonts

Kernel		ADK
Arial (Subset 1_30)		X
Courier New (Subset 1_30)		X
Symbol		X
Tahoma (Subset 1_30)		X
Tahoma Bold		X
Wingding		X

6.8 Graphics and Multimedia Technologies

Kernel		ADK
Audio		
Audio Compression Manager		X
Waveform Audio		X
Graphics		
Direct3D Mobile		x

DirectDraw		X
Gradient Fill Support		X
Still Image Codec Support (Encode and Decode)		X
Still Image Encoders and Decoders		X
BMP Decoder		X
GIF Decoder		X
JPG Decoder		X
PNG Decoder		X
Media		
Audio Codecs and Renderers		
MP3 Codec		X
MPEG-1 Layer 1 and 2 Audio Codec		X
MS ADPCM Audio Codec		X
Wave/AIFF/au/snd File Parser		X
Waveform Audio Renderer		X
WMA Codec		X
Direct Show		
ACM Wrapper Filter		X
DirectShow Core		X
DirectShow Display		X
DirectShow Error Messages		X
DirectShow Vide Capture		X
DMO Wrapper Filter		X
Media Formats		X
MPEG-1 Parser/Splitter		X
Video Codecs and Renderers		
DirectShow Video Renderer		X
MPEG-1 Video Codec		X
MS RLE Video Codec		X
Video/Image Compression Manager		X
WMV/MPEG-4 Video Codec		X
Windows Media Player		
Windows Media Player		X
Windows Media Player OCX		X
Windows Media Technologies		X

6.9 International

Kernel		ADK
Input Method Manager (IMM)		X
Local Services		
English (US) National Language Support only		-
National Language Support (NLS)		X

6.10 Internet Client Services

Kernel		ADK
Internet Explorer 6.0 for Windows Embedded CE – Standard Components		X
Internet Explorer 6.0 Sample Browser		X
Internet Explorer 6.0 for Windows Embedded CE Components		X
Internet Explorer Browser Control Host		X
Internet Explorer HTML/DHTML API		X

Internet Explorer Multiple-Language Base API		X
URL Moniker Services		X
Windows Internet Services		X
Internet Options Control Panel		X
Jscript 5.6		X
VBScript 5.6		X

6 .11 Security

Kernel		ADK
Authentication Services		X
NTLM		-
Schannel (SSL/TLS)		X
Credential Manager		X
Cryptography Services (CryptoAPI 1.0) with High Encryption Provider		X
Certificates (CryptoAPI 2.0)		X

6 .12 Shell and User Interface

Kernel		ADK
Graphics, Windowing and Events		
Minimal GDI Configuration		X
Minimal GWES Configuration		X
Minimal Input Configuration		X
Minimal Window Manager Configuration		X
Command Shell		
Aygshell API Set		X
Command Processor		X
Console Window		X
Graphical Shell		
Standard Shell		X
Common Controls		
Common Control		X
Common Dialog Support		X
Control Panel Applets		X
Mouse		X
Network User Interface		X
Software-based Input Panel (SIP)		X
SIP for Small Screens		X
Software-based Input Panel Driver		X
Touch Screen (Stylus)		X

6 .13 Device Driver

Kernel		ADK
Audio Driver for 7723SIU/ TLV320AIC23		X
NULL Display Driver (Stub)		-
Display Driver (LDCDC_SH7723)		X
TFT Res. 640x480, 320x240, col. 16Bit		X
Native keyboard support		X
Ethernet DM9000B		X
Asynctmac NDIS Driver		X

Serial interfaces (SCIF + SCIFA SH7723)		X
I2C (IIC SH7723)		X
Storage Devices		X
NAND Flash File System		X
Touch Driver (TSC2007)		X
USB Function with ISP1181B		X
USB Function Client Serial		X
USB Host (SH7723)		X

7 Application development

Application development is only possible with VS2005. Developing with Embedded Visual C++ is not supported with Windows Embedded CE 6.0 kernel. If there is a need for this, please don't hesitate to contact emtrion GmbH.

VS2005 supports unmanaged and managed code and programming languages like C++, C#, Visual Basic and more are available. The communication interface for debugging can be Ethernet as well as USB ActiveSync. Sole Ethernet communication without a little help from USB ActiveSync is possible, but needs some special tools instead. For a more detailed description see the next chapters. We recommend a combination of Ethernet and USBF and all the kernels are supporting this.

7.1 VS2005 for Application Development

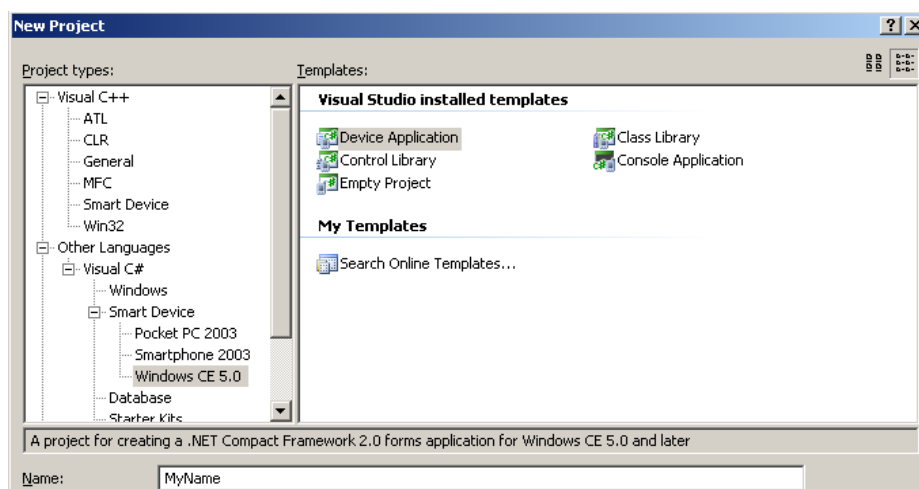
When you using VS2005 for application development based on a specific kernel some hints shall be considered.

- Since Win32-API functions may be omitted in the creation of operating system kernels under Windows® Embedded CE, there is no SDK (Software Development Kit) which suits all Windows® CE operating system kernels. Actually, each operating system kernel has an SDK of its own which must be installed in addition to the development environment.
- Please consult the online help of VS2005 (search for "Requirements") to obtain information on whether a specific function is available in an SDK. If the name of the OS kernel (e.g. DIMM-7723DevKit_CE600_SDK for starter kit kernel) is listed, the function will be supported by this OS kernel.
- The list of parameters of the Win32-API functions is identical with that of the corresponding API functions of Windows® 9x/NT/XP. With some functions certain parameters must be set to 0. For more information, please refer to the online help.

7.1.1 Creating a new Managed Project

The first step is starting a new instance of VS2005.

1. Select **File/NewProject...** from the Visual Studio menu.
2. In the **NewProject** window select **Visual C#/SmartDevice/WindowsCE5.0**
3. Select the **Device Application** template
4. Name your project **MyName** and click **ok**.



Now you can start with your application development.

7.1.2 Deploying to the target

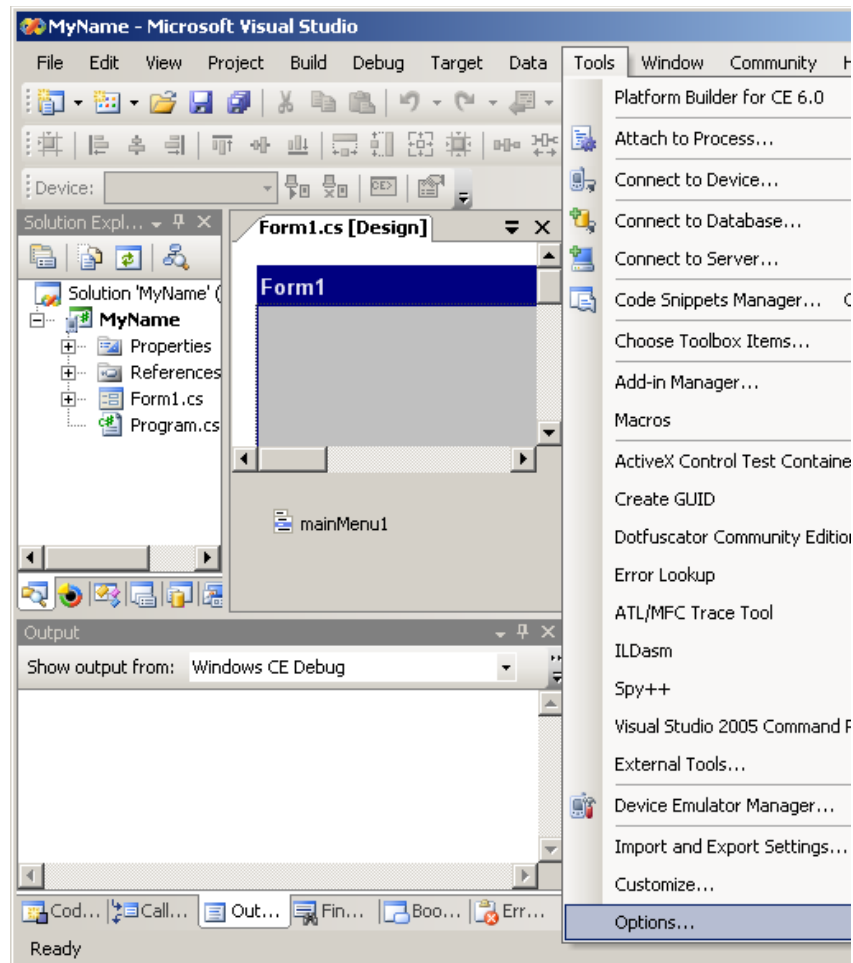
The connection between VS2005 and the target is made by network. For this reason the IP address of the target is necessary. To make the IP address known to VS2005 there are two options.

The first option is to ask manually by means of the function „ipconfig“ on the device. For that „ipconfig“ must be executed from the command shell.

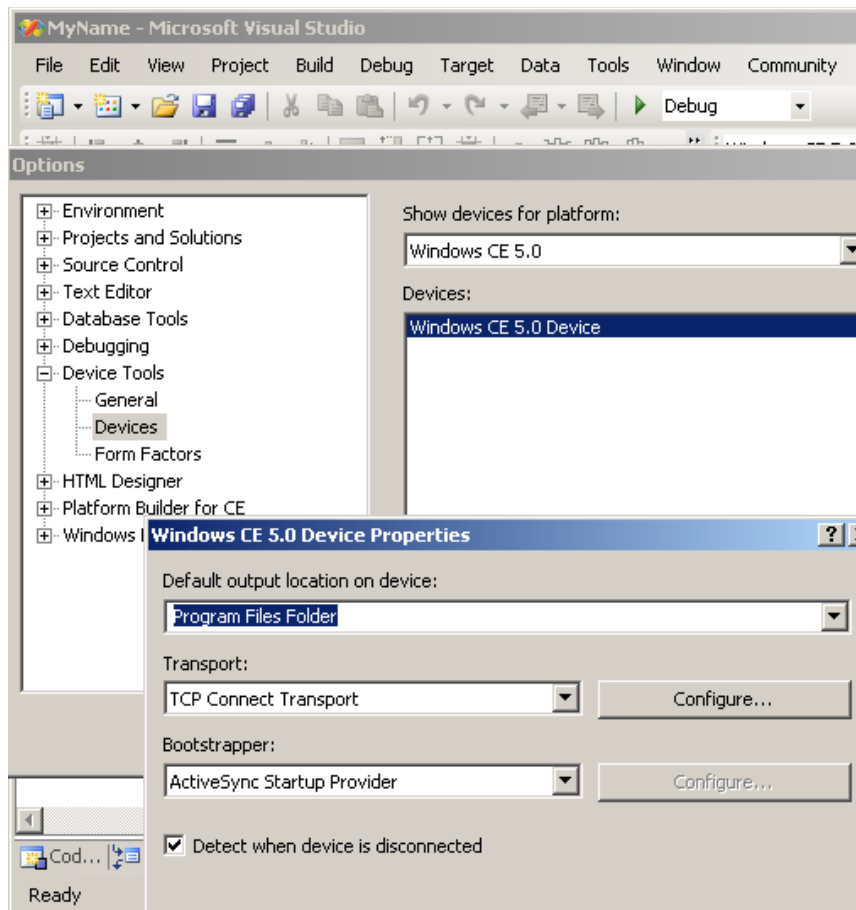
The second is receiving the IP address automatically by an existing USB ActiveSync connection.

After getting this address, the next step is preparing the managed application development environment for deployment.

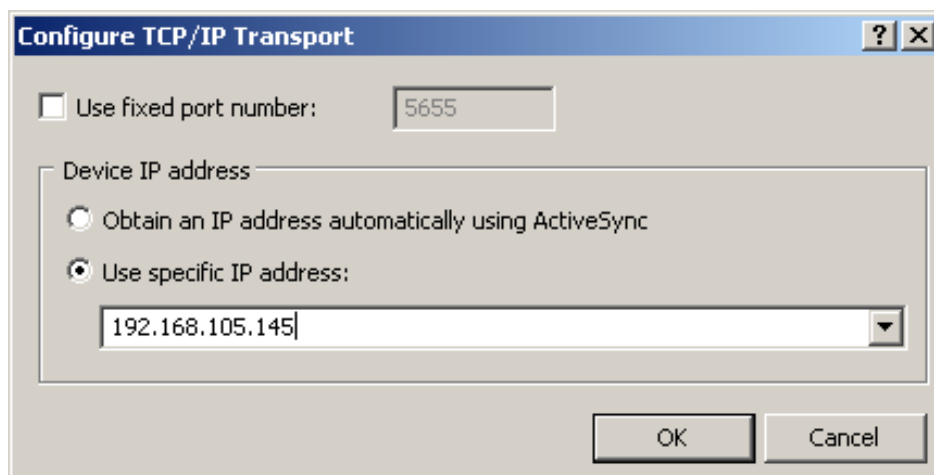
1. In VS2005 select **Tools/Options** from the menu.



2. In the options window, expand the **Device Tools** node and select **Devices**.
3. In the **Show device for platform:** drop down box select **WindowsCE5.0 Device**.
4. Click on **WindowsCE5.0 Device** and select **Properties**.



5. Click the **Configure** button beside the **Transport** drop down box. We are going to configure the TCP Connect Transport.



6. In the case when an USB ActiveSync connection exists between the workstation and the device, the option "Obtain an IP address ... is to select. In the other case the option "Use specific IP address" is to select and the IP address you got on the device by ipconfig has to be inserted.
7. Click **OK** through all of the dialogs.

Preparing the target:

1. At the command shell, type **ConmanClient2**.
2. Then, type **cmaccept**. You have 3 minutes to establish a connection with your managed application.

Deploying the managed application:

1. Select **Debug/Start Debugging** from VS2005.
2. Select **WindowsCE5.0 Device** from the list of devices in the **Deploy MyName** box and click **Deploy**. VS2005 will deploy several cab files to the device in addition to your application. Your application is executed on the target device.

8 Starting an application at system start

There are two options to start an application at start up. Using only one option is allowed at any time. In any case, the execution of the explorer is controlled by emtrion's autostart mechanism described in the section below. This means no launch entry may be existed in the registry for the explorer.

To use the autostart functionality the registry must be modified. For it you can use the remote Registry Editor of VS2005. Making the changes persistent the emtrion tool WriteReg.exe is available.

Additional items have to be considered when you want to realise starting your application at system start.

- Folder structures and shortcuts get lost at power off or reset when it have been created at runtime.
- The same as above happens with files that have been copied to the object store of the kernel.
- Changes to the registry also getting lost when it have not been made persistent.
- Due to supporting autostart of applications from a storage device, the kernel is configured with search paths of any supported storage device.

Dependent on the supported storage devices of the Developer Kit the storage folders are specified as following.

- ...\\SD Card for storage device microSD or SD Card
- ...\\MultiMediaCard for storage device MMC
- ...\\Hard Disk for storage device USB stick
- ...\\FlashDisk for storage device flash
- ...\\NAND Flash for storage device nand

8.1 Emtrion's autostart mechanism

One possibility to start your application at system start is using the autostart mechanism from emtrion. For them an autostart program was written and included into the kernel.

The registry of the Developer Kit kernel includes a launch link to that autostart program and starting it at system start. Additional, a shortcut of the autostart program is located in the "StartUp" folder of the explorer, so that the start up mechanism of the explorer is also useable. Now, the autostart program itself looks in a specific registry key whether or not what application shall be started. The registry key includes several entries.

But this is not the only task of the autostart program. The autostart program is also controlling the starting of the explorer.

The autostart program looks in **[HKEY_LOCAL_MACHINE\\Software\\emtrion\\AutoStart]** for the entry **"CustomerApp"**. The entry is of type REG_SZ and is provided for the filename and if necessary the path of the application that shall be started. When the entry is empty no further action will be done. In the case of a valid entry the autostart program tries to start the application. When a path is not stated, the system searches in the folder "Windows" and in the folders of the supported storage devices of the Developer Kit, mentioned in the introduction of the chapter.

The entry **"AppParameter"** can be found in the same registry key. The entry is also a type of REG_SZ and is provided for the command line parameters of the application.

Optionally, the same registry key may also contain an entry called **"WaitCycles"**. This entry, a type of REG_DWORD, lets you extend the time that the autostart program is waiting for the directories to be linked, e.g. a FlashDisk. If the entry is not found, a default value of 100 is taken.

Optionally, the same registry key may also contain an entry called **"FailureOptions"**. This entry is also a type of REG_DWORD and is bit-coded. Please set only these bits that are really necessary. The meaning of the individual bits is shown in the table below.

Bit No.	meaning
0	Setting this bit, a message is displaying in a message box, when the custom application cannot be started.
1	With this bit you can specify the message showing in the message box. '0': message → "Cannot start the application which was specified to launch at startup." is showing. '1': → more detailed information is showing, like the name and search path of the application.
2	Setting this bit, the autostart program doesn't exit and is showing continuously the message when the custom application cannot be started.
3	Setting this bit, the explorer is started when the custom application cannot be started.
4	Specifies the start up process for the custom application 0: → the explorer is not started and the custom application is started at launch time of the autostart program. 1: → the explorer is started and the custom application is started by the startup mechanism of the explorer
5	Setting this bit, the explorer will be started in any case.
6	Setting this bit when the kernel is a headless one. Due to no graphic exists, the console is started instead.
7..31	Reserved for future extensions

A further entry is "**Timeout**". On that you can specify the time between two consecutive WaitCycles. The type of Timeout is REG_DWORD.

8.2 Autostart with launchXX and dependXX

This section describes how you can modify the kernel in order to start your application at system start by the two registry entries launchXX and dependXX.

Using this option, be sure the registry key "**CustomerApp**" in [Emtrion's autostart mechanism](#) is empty. In the case of using the explorer by the kernel, the registry key "**FailureOptions**" has to be set to 00000020h for execution.

The application and its own DLLs that shall be started have to be stored in one of the supported storage devices.

- Here are the settings to do on the registry key [HKEY_LOCAL_MACHINE\init]:
 - Entry 1:
 - Name: LaunchXX, where XX stands for a two-digit number. This number must be between 51 and 99 with the operating system kernel belonging to the starter kit kernel.
 - Type: REG_SZ
 - Value: Name of the file that shall be executed
 - Entry 2:
 - Name: DependXX, where XX must be the **same** two-digit number as specified with entry 1.
 - Type: REG_BINARY
 - Value: Hexadecimal digits that specify the dependency on a previously started program.

Please also note section "Persistent Registry".

Note

Entry 2 specifies the dependency of the application. An application stored on a mass storage cannot be started until the corresponding entry has been made in the folder structure. It may happen in this context that the registry entries in the [HKEY_LOCAL_MACHINE\init] key will be processed faster than the mass storages are entered in the folder structure. This must be taken into account in the start sequence.

9 Data Exchange between Target and Workstation

The Developer Kit kernel allows the exchange of data between the target and your workstation. For this, Microsoft's ActiveSync program is needed. The latest version is available on the internet at:

<http://search.microsoft.com/search/results.aspx?st=b&qu=ActiveSync&view=en-us>

If you have not yet worked with Microsoft's ActiveSync, we recommend to you a look at section [#Remarks on Using ActiveSync](#).

9.1 Preparing the Desktop Computer

If Microsoft ActiveSync has not yet been installed on your desktop computer, it must be prepared as follows:

Windows 2000:

1. From the Control panel choose "Modem" and install the communications cable between two computers.
2. Start **ActiveSync_45_eng_setup.exe** or **ActiveSync_45_ger_setup.exe**. Both have been placed in the subdirectory **ActiveSync** of the Starter Kit CD.

With all other Windows versions:

1. Start **ActiveSync_45_eng_setup.exe** or **ActiveSync_45_ger_setup.exe** in the **ActiveSync** directory of the starter kit CD.

9.2 Preparing the Windows CE Device

USB function is the default setting for communication with ActiveSync. This communication is sufficient for file transfer and for application debugging.

When you do application development by VS2005 communication via Ethernet is also possible. In this case USB ActiveSync can be used to get the ip-address of the device.

9.3 USB ActiveSync

- Using a USB cable to connect the device to the workstation.
- When not yet powered on, turn on the power on the target. The Windows® CE OS kernel which is stored in the flash will be executed. The running kernel will initiate an ActiveSync connection at the desktop computer. When the target is unknown as USB-Device at the workstation, ActiveSync will ask for the corresponding USB-function-driver which has to be installed. The driver is located in the subdirectory **USBFDriver** of the installation.
- When ActiveSync does not start automatically on the workstation then you must perform this manually. Please note, USB has to be selected in the connectivity settings of ActiveSync

The target establishes a connection with the desktop computer and possibly prompts you to login and enter your password. Enter the same name and password as you are using for your desktop computer.

Note :


When the connection is **not** automatically established between the target and desktop, you should start the "repllog" program on the target manually. To do this, select "Run" from the "Start" menu, then enter **repllog** and click the **OK** button.

The **New Partnership** dialog appears on the desktop computer.

- Select the options as described in section [#Remarks on Using ActiveSync](#).
- After establishing a connection, click **Explore** to open a window. Via this window, you may then exchange files with the target using drag and drop. For this, drag the files onto the desired directory on the target.

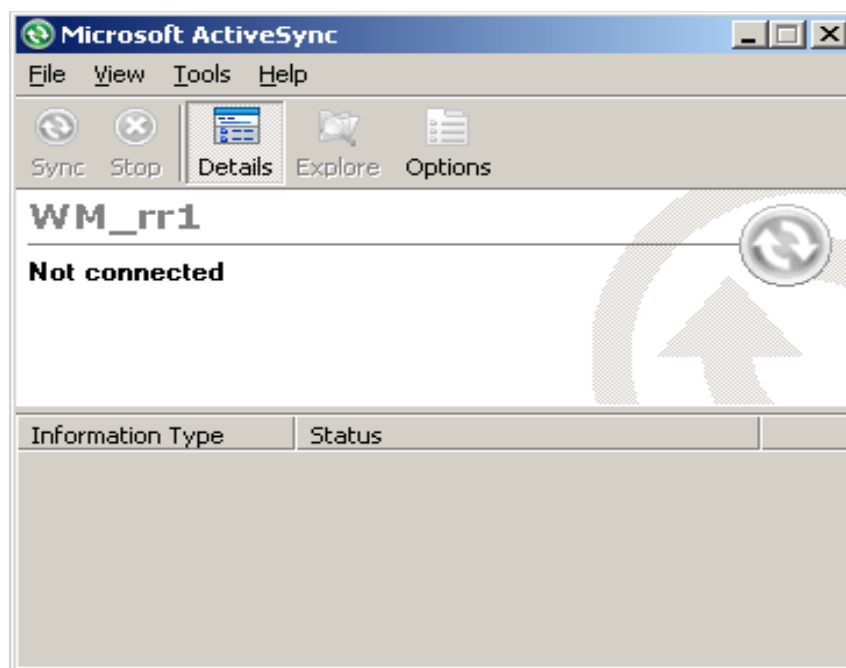
Note

The folders of the kernel and their contents are created when Windows® CE is starting. If you copy any files to these folders, they are not available any more when Windows® CE is restarted again. This does not apply to folders and files which are directly located at mass storages like the “..\FlashDisk” or the “..\SD Card”. These folders are available only if the corresponding devices are detected in the system during the start up phase. The subfolders of the “Network” folder are a special case. These folders represent drives when a connection via network was established.

To terminate the connection click the  symbol on the Windows® CE's task bar. The connection can be detached in the dialog that appears. Another easy way to stop ActiveSync is just to unplug the USB cable.

9.4 Remarks on Using ActiveSync

After installing ActiveSync, there is a "Microsoft ActiveSync" link in the "Programs" group of the Windows "Start" menu. When pointing to this link, the following window appears:



Please verify that the USB connection is enabled (**Connection Settings** menu item of the **File** menu). If a connection has been established for the first time after power on, the following dialog box displays:



The window shows that no partnership has been established between the Windows® CE computer and the desktop PC.

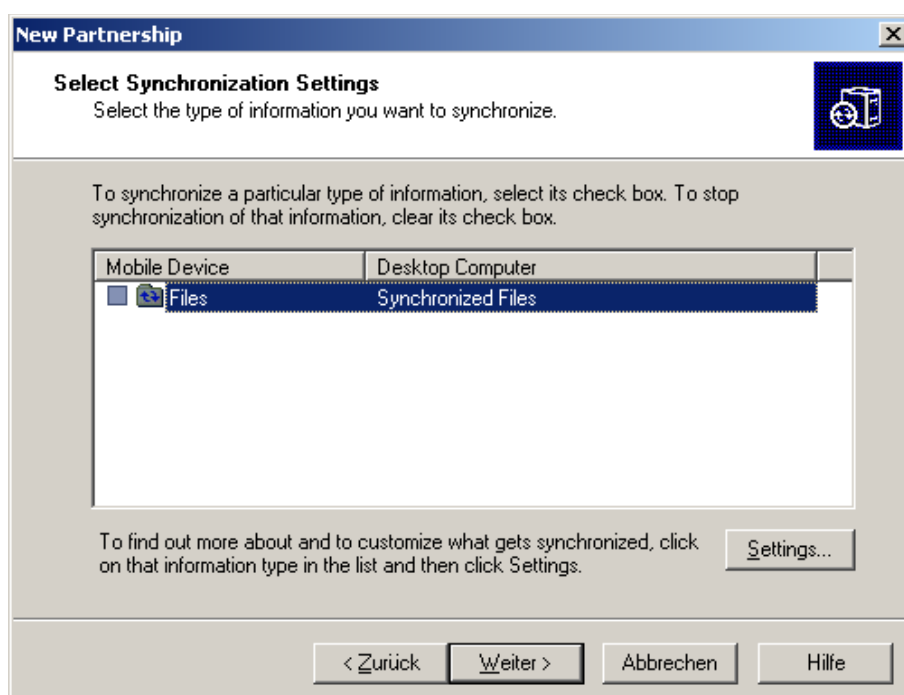
Hint

If a partnership has already been created with a device with persistent registry, the dialogs displayed here do not show up.

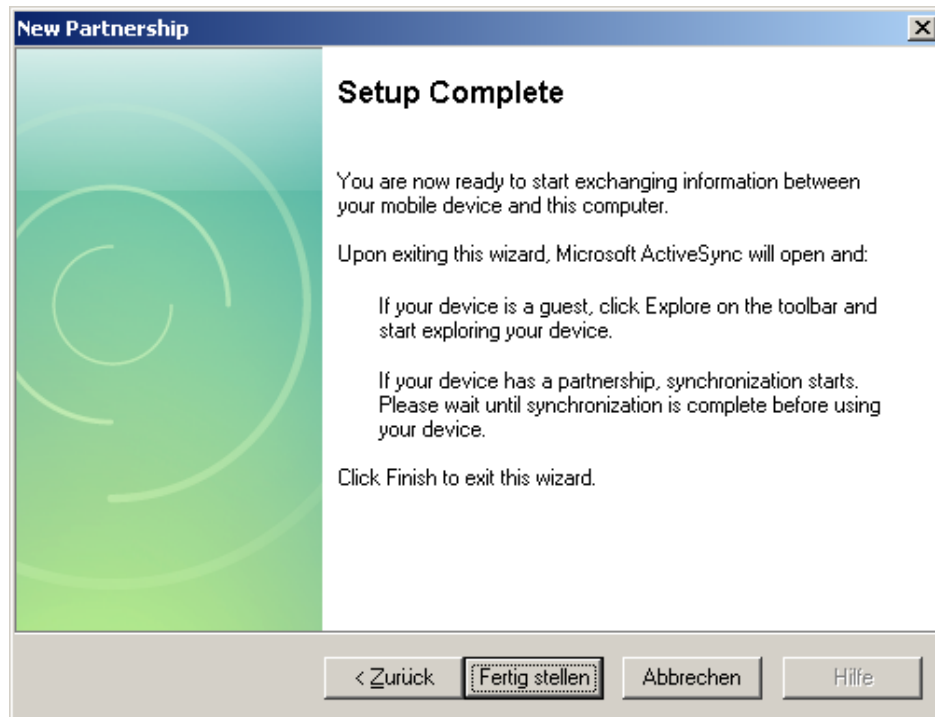
Any of the offered options will be using the USB interface with following property.

"**No**" means that the Windows® CE computer is connected with the desktop computer as a guest only. The connection as a guest is sufficient if you wish to exchange data or to debug an application via an USB connection. "**Yes**" will establish a partnership, which is necessary for data synchronisation.

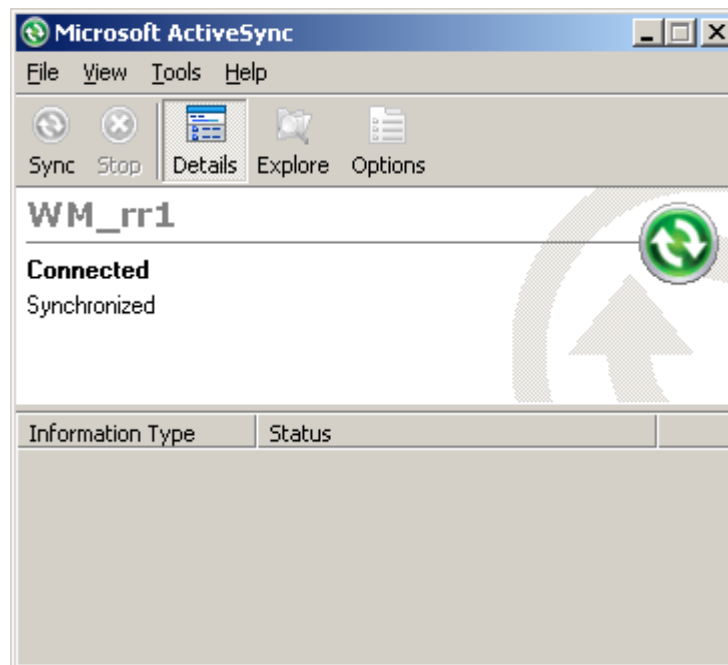
The following dialogs appear.



Accept the default setting or activate "Files" by clicking on the small box left of "Files" and click "Next".



After that, a dialog box appears telling you that the partnership has been successfully established. After clicking the **Finish** button, the ActiveSync program will open on your desktop:



The connection has been established.

Data transfer

Clicking the "Explore" button will bring up a window where you can search the Windows® CE computer.

The data exchange between the Windows® CE computer and the desktop computer can be initiated by dragging a file from the window to another window that does not belong to ActiveSync, or vice versa.

10 Multimedia sample applications

To demonstrate the excellent multimedia characteristics of the SH7723 two sample applications have been included into the image. Please consider that both applications require the HiCO.DIMM-Base.

10 .1 UMP_600_V2.exe

This application is playing decoded video files of several formats. What formats exactly and how to serve the application is described in the manual "UMP_User_Developer_Documentation...".

10 .2 Videointest.exe

This application is capturing of video input like PAL, NTSC or from digital interface. The video input can be select via command line parameter. Default is set PAL. By a second command line parameter you can demonstrate the influence of blending by HW or by SW. More details about the application are described in section 4. of the "CEU_Driver_User_Documentation...".

11 Persistent Registry

By default, Microsoft's Windows® Embedded CE does not support a persistent registry. That is, all modifications made to the registry while Windows® CE is running will not be kept when Windows® CE is restarted.

To permanently keep these modifications, two functions are available which have to be implemented by the hardware manufacturer. These functions are already implemented in the kernel. They will be called by the operating system, in order to write the registry onto the Flash in the last 512 kBytes.

Note

Systems with a non-persistent registry have the advantage that a system cannot be damaged due to possibly wrong registry settings made while the system is running.

In order to have the registry write to the flash disk, the "RegFlushKey" function must be called by the application. However, this function should not be called after each modification to avoid performance bottlenecks. Call this function after having made several changes to the registry.

The Developer Kit's operating system kernels include a tool called WriteReg.exe. This tool calls the RegFlushKey function to make the desired changes. For example, it can be used for system modifications that are to be kept permanently.

"writereg -default" sets the registry to its default values, this means that all changes are lost and the settings of the default registry are used.

A programming sample:

```
HKEY    hRegKey;
DWORD   retWert;

/* open the registry key */
retWert = RegOpenKeyEx(HKEY_LOCAL_MACHINE, T("Software"), 0, 0, &hRegKey);

if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/*
:
:
Here new entries are made, entries changed or deleted.
:
:
*/

/* Save registry to Flash */
retWert = RegFlushKey ( hRegKey );
/* Hint: Here a valid handle for the registry key will need to
be specified (see the online help).
The complete registry will always be saved to Flash. */
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/* Close registry key */
retWert = RegCloseKey( hRegKey );
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}
```


11.1 Deleting the Persistent Registry

The persistent registry can be deleted in either of the following ways:

- 1.) Using the bootloader
- 2.) Via an application under Windows CE

11.1.1 Deleting the Registry by the Bootloader

The persistent registry can be deleted via menu item 2 of the extended bootloader menu described in section [Bootmenu](#).

The persistent registry can be deleted via menu item 4 of the second bootloader menu. For more detailed information, please refer to the user manual of the bootloader.

11.1.2 Deleting the Registry by means of an Application under Windows CE

IMPORTANT: This works only with images which contains the RAM-based registry. For images with Hive-based registry you have to use the bootloader to clear the registry.

A Windows CE application can be enabled to use the persistent registry by calling the Win32-API function `KernelloControl`. For this purpose, emtrion has created the following device-specific I/O control code:

```
#define IOCTL_HAL_SETREGISTRYTODEFAULT CTL_CODE(FILE_DEVICE_HAL, 0x810,
METHOD_BUFFERED, FILE_ANY_ACCESS )
```

In this code, `CTL_CODE` is a macro, `FILE_DEVICE_HAL`, `METHOD_BUFFERED` and `FILE_ANY_ACCESS` are specified as `#define` statements. Both the macro and the definitions were defined by Microsoft in the `winioctl.h` header file. This file is usually linked to applications with the `#include <windows.h>` statement.

When you want to learn more details about `KernelloControl` we refer to the online help of the Platform Builder.

IOCTL_HAL_SETREGISTRYTODEFAULT

This control code will delete the currently stored persistent registry. As a result, the default registry will be used when the system is started again. However, a reset will **not** be issued automatically. Such a reset can be performed by using the `IOCTL_HAL_REBOOT` control code.

`IOCTL_HAL_SETREGISTRYTODEFAULT` expects the following parameters of the `KernelloControl` function:

<code>lpInBuf, nInBufSize, lpOutBuf, nOutBufSize</code>	These parameters will not be analyzed
<code>lpBytesReturned</code>	Pointer to a <code>DWORD</code> variable containing the number of bytes returned (here 0)

The control code deletes the flash sectors (where the registry is saved) by means of the erase function supplied by the bootloader.

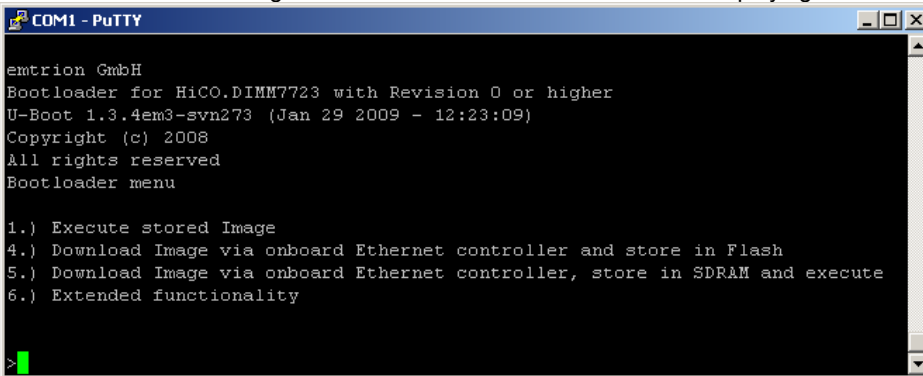
12 Downloading the image to the target

The images generated for DIMM-SH7723 are uncompressed and equipped with a special header. Furthermore, the name of the image has to be the same as the value of the environment variable “wimg” of the bootloader. By delivery “wimg” is set to “wce-hico7723”. Without changing of “wimg” rename the name of the image to “wce-hico7723, when not set yet. When this condition is achieved the image can be downloaded through TFTP by the menu of the bootloader.

12.1 Perform downloading

We assume the target and host have been set up as described in [Setting up the target](#) and [Setting up the host](#).

- Make sure a TFTP server is running and its base directory is pointed to the location of the extended wince image.
- Power ON or reset the target so that the menu of the bootloader is displaying at the terminal.



```
COM1 - PuTTY

emtrion GmbH
Bootloader for HiCO.DIMM7723 with Revision 0 or higher
U-Boot 1.3.4em3-svn273 (Jan 29 2009 - 12:23:09)
Copyright (c) 2008
All rights reserved
Bootloader menu

1.) Execute stored Image
4.) Download Image via onboard Ethernet controller and store in Flash
5.) Download Image via onboard Ethernet controller, store in SDRAM and execute
6.) Extended functionality

>
```

- Clean the Persistent Registry within the item “Extended functionality”
- Choose item “4” or “5” to start downloading of the image either to Flash or SDRAM

13 The Bootloader

This section gives a briefly description of the bootloader used in this Developer Kit. When you are more interested in the function scope of the bootloader, please refer to the bootloader manual.

The main task of the bootloader is to download a WINCE-Image to the target and starting it. To achieve this work the bootloader can be serviced through a terminal running on a host. For it, the host has to be connected to the serial port UART A at the target.

13.1 Location of the bootloader

The bootloader is located in the lower address area of the NOR-Flash and starts running at power ON or a reset. At this point the bootloader is evaluating the environment variable “bootcmd”, which is set to the command bootmenu. However the work of the bootloader is dependent on the mode he is configured.

13.2 Communication settings

The bootloader's communication settings are.

- Baudrate: 115200bps
- Data bits: 8
- Stop bits: 1
- Parity none
- Handshake none

13.3 Console mode

In console mode the bootmenu is putting out via the serial port UART A. To configure the bootloader for using the console, the switch SW1-4 has to be turned off on DIMM-SH7723

DIP Switch SW1 for console mode:

	4	3	2	1
off	X	X	X	X
on				

13.4 Boot mode

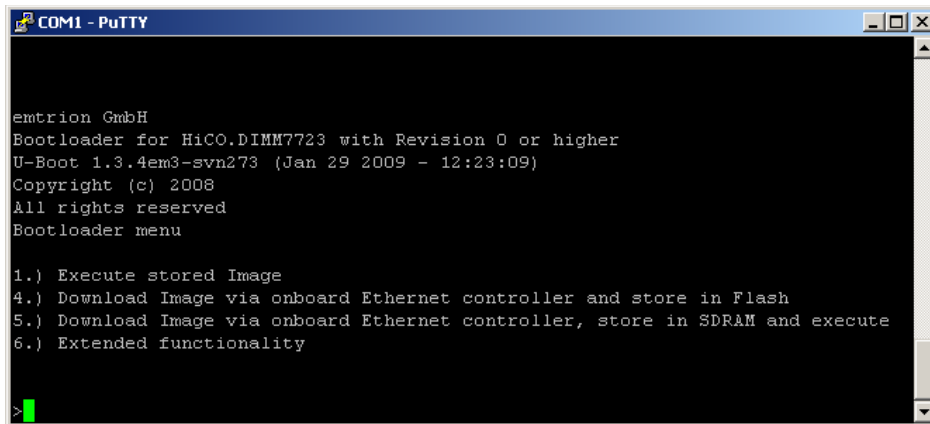
In boot mode the bootloader doesn't make any output to the serial port UART A. But due to the boot mode, item 1 of the bootmenu is launched and the stored image is executed. The boot mode is active when switch SW1-4 is turned on.

DIP Switch SW1 for boot mode:

	4	3	2	1
off		X	X	X
on	X			

13.5 Bootmenu

When the bootloader is set for console and the target is connected to the host where a terminal is running, the bootloader is putting out a menu after power ON or a reset.



```
COM1 - PuTTY

emtrion GmbH
Bootloader for HiCO.DIMM723 with Revision 0 or higher
U-Boot 1.3.4em3-svn273 (Jan 29 2009 - 12:23:09)
Copyright (c) 2008
All rights reserved
Bootloader menu

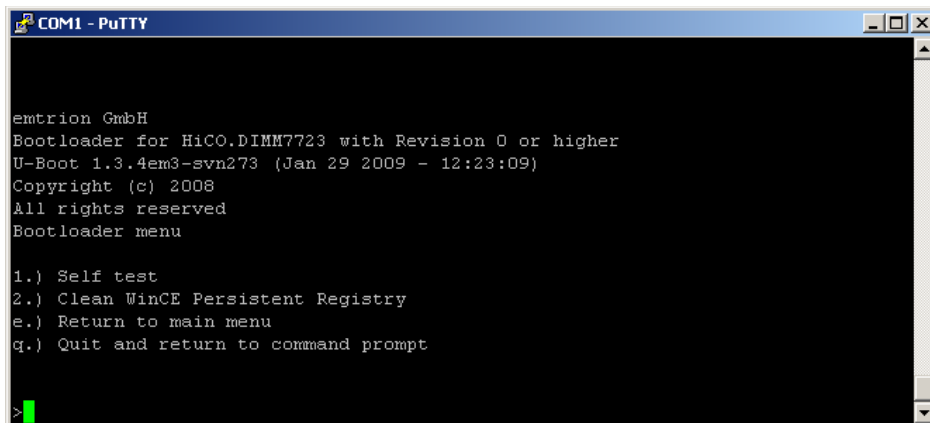
1.) Execute stored Image
4.) Download Image via onboard Ethernet controller and store in Flash
5.) Download Image via onboard Ethernet controller, store in SDRAM and execute
6.) Extended functionality

>
```

With item “1” the currently stored image can be executed manually. The image will be first copied from NAND to RAM and is then started.

With item “4” and “5” an image can be downloaded via tftp and either stored to the NAND or to the RAM. The name of the image is defined by the environment variable “wimg” of the bootloader. The name of the image for this Developer Kit is set to “wce-hico7723”.

Choose item “6” the extended menu gets open.



```
COM1 - PuTTY

emtrion GmbH
Bootloader for HiCO.DIMM723 with Revision 0 or higher
U-Boot 1.3.4em3-svn273 (Jan 29 2009 - 12:23:09)
Copyright (c) 2008
All rights reserved
Bootloader menu

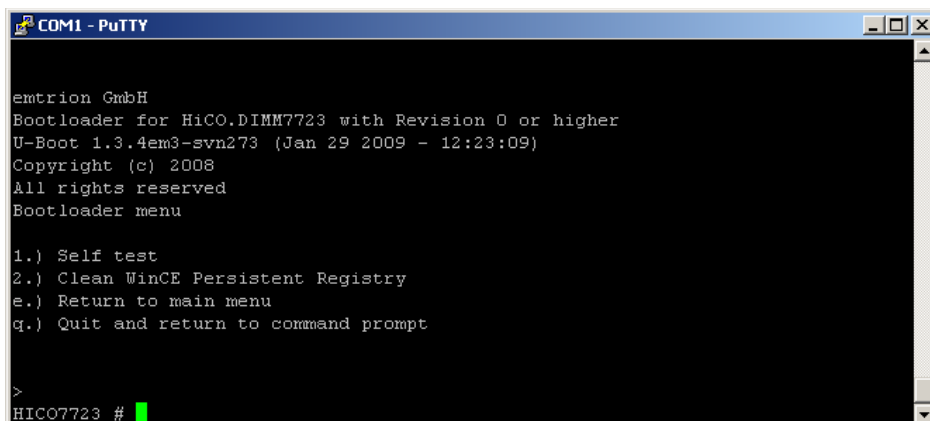
1.) Self test
2.) Clean WinCE Persistent Registry
e.) Return to main menu
q.) Quit and return to command prompt

>
```

With item “2” you can manually clean the Persistent Registry of WINCE. This should always be done at any time before downloading the WINCE-Image.

13.6 Get to the bootloader prompt

From the second menu you can change to the bootloader prompt by choosing the item “q”.



```
COM1 - PuTTY

emtrion GmbH
Bootloader for HiCO.DIMM723 with Revision 0 or higher
U-Boot 1.3.4em3-svn273 (Jan 29 2009 - 12:23:09)
Copyright (c) 2008
All rights reserved
Bootloader menu

1.) Self test
2.) Clean WinCE Persistent Registry
e.) Return to main menu
q.) Quit and return to command prompt

>
HICO7723 #
```

At this prompt level, you can

- executing all the commands of the bootloader
- looking how the environment variables of the bootloader are defined to by the command “printenv”
- setting environment variables like for networking, booting and so on

To get back to the menu of the bootloader, enter the command “bootmenu” at the prompt.

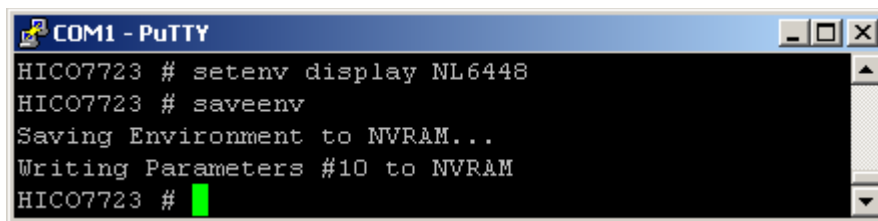
13.7 Changing the display

The kernel just supports two displays.

Display	Resolution	Colours	Remarks
TFT, NEC NL6448BC20	640x480	16bpp	Name for display NL6448
TFT, TX14D11VM1CBA	320x240	16bpp	Name for display TX14

To offer a wide flexibility the used display has to be selected in the bootloader. For this purpose the bootloader holds the environment variable “display”. Set the variable to the corresponding name of the used display and then save the environment variable. However, the environment variable is already set to the connected display by delivery.

The following example shows the setting for the display NEC NL6448BC20 on the bootloader output. HICO7723 # presents the prompt of the bootloader. To get to the bootloader prompt you have to leave the bootloader menu. For more information please refer to the chapter [The Bootloader](#).



```

COM1 - PuTTY
HICO7723 # setenv display NL6448
HICO7723 # saveenv
Saving Environment to NVRAM...
Writing Parameters #10 to NVRAM
HICO7723 # █

```

Anytime you change the display, the touch has to be recalibrated. This can be done with the calibration applet “Stylus” in the control panel. After this make the new calibration data persistent by using the tool writereg.exe.

13.8 Supported File Formats

The bootloader requires a special file. It consists of the raw binary nk.nb0 extended by a special header. The special header contains all the information needed by the bootloader, like length, start address etc.

14 Serial Ports

The SH7723 has two three-channel serial communication interfaces with FIFO, SCIF and SCIFA. Two channels of the interface SCIF and all channels of the SCIFA are supported in asynchronous mode. That makes a total of 5 serial interfaces available.

The relationship between the names and serial ports from the hardware and software point of view and some more supported features are summarised in the table below.

Software	COM4	COM5	COM1	COM2	COM3
Channel/ interface	SCIF1/ SCIF	SCIF2/ SCIF	SCIF3/ SCIFA	SCIF4/ SCIFA	SCIF5/ SCIFA
Signals	LVTTL*	LVTTL*	RS232	LVTTL*	LVTTL*
HiCO.DIMM-Base UART-x Connector	UART-E pin header J15 **	UART-D pin header J15 **	UART-A D-Sub (bottom)	UART-C pin header J15 **	UART-B D-Sub (top)
HiCO.DIMM- ECOBBase UART-x Connector on	UART-E pin header J12 **	UART-D pin header J12 **	UART-A D-Sub	UART-B pin header J12 **	UART-C pin header J12 **
HW-Handshake RTS/CTS	No	No	Yes	No	No
Hardware Flow Control RTS/CTS	No	No	No	No	No
HW-FiFo - Receive - Transmit	16Byte 16Byte	16Byte 16Byte	64Byte 64Byte	64Byte 64Byte	64Byte 64Byte
Baudrates - 1200 - 2400 - 4800 - 9600 - 14400 - 19200 - 38400 - 57600 - 115200 - > 115200	yes yes yes yes yes yes yes yes yes ***	yes yes yes yes yes yes yes yes yes ***	yes yes yes yes yes yes yes yes yes ***	yes yes yes yes yes yes yes yes yes ***	yes yes yes yes yes yes yes yes yes ***

(*) Suitable RS232-adapters are available by emtrion

(**) For more details please refer to the corresponding HW- manual

(***) Only on request

All the serial ports can be accessed by the WIN32-API of WINCE. But with the debug version of the Developer Kit kernel, **COM1** is reserved for debug purpose.

15 I²C Bus Driver

15.1 General Information

The processor module DIMM-SH7723 has one I²C bus. The bus is available on the HiCO.DIMM connector. Dependent on the base board from emtrion, the bus is routed to J17 on the HiCO.DIMM-Base or to J4 on the HiCO.DIMM-ECOBBase.

The I2C-bus is running at 100KHz and several devices are already connected to it, such as RTC, touch controller and the audio codec. Their device addresses are 0x68, 0x48 and 0x1b. They are serviced by the kernel's I2C interface. For supporting additional devices the kernel's I2C interface is extended to a Stream Interface one.

15.2 Implementation

The driver has been implemented as Stream Interface driver and is included to the kernel. The user is thus able to serve the interface, similar to a serial interface, with the Win32-API functions CreateFile, ReadFile, WriteFile, CloseHandle and DeviceIoControl. As a prefix for the interface, "IIC" is used instead of "COM".

The interface IIC0: may be opened several times. This is useful when several components on the I²C bus are to be accessed. The application thus gets various handles on the interface, which allows the access of different devices.

15.3 API

15.3.1 CreateFile

Create and return a handle for I2C driver

Proto:

```
HANDLE CreateFile(
    TEXT("IIC0:"),
    GENERIC_READ | GENERIC_WRITE,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL
);
```

Parameters:

Return value:

A handle of IIC driver is returned if the function is successfully processed; otherwise, NULL is returned. To obtain extended error information, call GetLastError function.

15.3.2 DeviceloControl

The following IO Codes are supported by the driver.

IOCTL code	Value	Description
IOCTL_I2CBUS_SETADDRESS	08002000h	Sets the device address
IOCTL_I2CBUS_READWRITE	08002004h	Reads and writes to the specified device

15.3.2.1 IOCTL_I2CBUS_SETADDRESS

The API sets the device address to the open context of the specified driver handle.

Proto:

```

BOOL DeviceloControl(
    hI2cDrv,
    IOCTL_I2CBUS_SETADDRESS,
    lpInBuf,
    6,
    NULL,
    0,
    NULL,
    NULL
);

```

Parameters:

hI2cDrv: IIC driver handle created by CreateFile

lpInBuf: Pointer to the buffer that passes the device address to the driver as follows.

Offset	Size	Description
0	2	Device address, only 7bit addresses allowed
2	4	Reserved for future use, set to 0

Return value:

If returns TRUE, the IOCTL was successfully processed; otherwise, returns FALSE. To obtain extended error information, call GetLastError function and look for identifying the error at the table below.

15.3.2.2 IOCTL_I2CBUS_READWRITE

The API performs either only writing or writing and reading to an IIC address. In the case of writing and reading the function is executed as “repeated start condition”.

Proto:

```
BOOL DeviceIoControl(
    _In_       HANDLE hI2cDrv,
    _In_       DWORD  IOCTL_I2CBUS_READWRITE,
    _In_       LPVOID lpInBuf,
    _In_       DWORD  inLen,
    _Out_      LPVOID lpOutBuf,
    _In_       DWORD  outLen,
    _Out_      LPDWORD lpBytesReturned,
    _In_       PVOID  NULL
);
```

Parameters:

hI2cDrv: IIC driver handle created by CreateFile

lpInBuf: The in buffer comprises the structure sI2CControlIn and passes the information to the driver as follows.

```
typedef struct _I2CControlIn {
    WORD        wAddress;
    DWORD       dwFlags;
    BYTE        bNumberOfBytesToRead;
    BYTE        bNumberOfBytesToWrite;
    BYTE        bData;
} sI2CControlIn, *psI2CControlIn;
```

Structure element	Description
wAddress	Device address, only 7bit addresses allowed
dwFlags	Reserved for future use
bNumberOfBytesToRead	Number of bytes to read
bNumberOfBytesToWrite	Number of bytes to write
bData	Start of room for the data to write through I2C. The amount of data must agree with bNumberOfBytesToWrite

inLen: Number of bytes of the whole in buffer: sizeof(sI2CControlIn) + bNumberOfBytesToWrite

lpOutBuf: out buffer that receives the data read from I2C.

outLen: Number of bytes to be read from I2C: → bNumberOfBytesToRead

lpBytesReturned: Pointer to the number of bytes read.

Due to determine the same size of the structure sI2CControlIn at application and driver level, the structure has to be compiled as **packed**.

Return value:

If returns TRUE, the IOCTL was successfully processed; otherwise, returns FALSE. To obtain extended error information, call GetLastError function and look for identifying the error at the table below.

15.3.3 ReadFile

ReadFile reads data through I2C. Before reading from the device, be sure the corresponding device address is set by IOCTL_I2CBUS_SETADDRESS.

Proto:

```
BOOL ReadFile(  
    hI2cDrv,  
    lpBuf,  
    nNumberOfBytesToRead,  
    lpNumberOfBytesRead,  
    NULL  
);
```

Parameters:

hI2cDrv: IIC driver handle created by CreateFile
lpBuf: Pointer to the buffer that receives the data read from I2C.
nNumberOfBytesToRead: Number of bytes to be read from I2C.
lpNumberOfBytesRead: Pointer to the number of bytes read.

Return value:

Nonzero indicates success. Zeros indicates failure. To get extended error information, call GetLastError function and look for identifying the error at the table below.

15.3.4 WriteFile

WriteFile writes data through I2C. Before writing to the device, be sure the corresponding device address has been set by IOCTL_I2CBUS_SETADDRESS first.

Proto:

```
BOOL WriteFile(  
    hI2cDrv,  
    lpBuf,  
    nNumberOfBytesToWrite,  
    lpNumberOfBytesWrite,  
    NULL  
);
```

Parameters:

hI2cDrv: IIC driver handle created by CreateFile
lpBuf: Pointer to the buffer containing the data to write through I2C.
nNumberOfBytesToWrite: Number of bytes to write through I2C.
lpNumberOfBytesRead: Pointer to the number of bytes written.

Return value:

Nonzero indicates success. Zeros indicates failure. To get extended error information, call GetLastError function and look for identifying the error at the table below.

15.3.5 CloseHandle

This API closes the I2C handle.

Proto:

```
BOOL CloseHandle(
    HANDLE hI2cDrv
);
```

Parameters:

hI2cDrv: IIC driver handle created by CreateFile

Return value:

Nonzero indicates success. Zeros indicates failure. To get extended error information, call GetLastError function and look for identifying the error at the table below.

15.3.6 Error Codes

In this section, the driver errors are listed that can be occurred during processing the API.

Error Code	Value	Description
ERROR_INVALID_HANDLE	ffffffffh	The handle hI2cDrv is not valid
ERROR_ACCESS_DENIED	00000005h	The handle has no access permission for reading or writing
ERROR_INVALID_PARAMETER	00000057h	Unknown device context
ERROR_NOT_ENOUGH_MEMORY	00000008h	Driver is out of memory or the structure elements inLen and outLen don't agree with the total length of the corresponding buffer
HTX_ERROR_INVALID_DEVICEADDRESS	20000001h	Device address is not set
HTX_ERROR_INVALID_FLAGS	20000002h	Internal driver flags are not set
HTX_ERROR_NORESPONSE	20000003h	Device is not responding to the master
HTX_ERROR_BUSBUSY	20000004h	the bus is used or affected by another device
HTX_ERROR_ERROR_RESPONSE	20000005h	Device is not responding while transferring data

Error Code	Value	Description
HTX_ERROR_UNKNOWN	20000006h	Occurred error is not known

16 Licence Agreements

SOFTWARE LICENSE TERMS

WINDOWS EMBEDDED CE 6.0 CUSTOMIZED OEM SOFTWARE DEVELOPMENT KIT for

Windows Embedded CE 6.0 Developer Kit for DIMM-SH7723, 02.0.0

These license terms are an agreement between you and [emtrion GmbH] "Company". Please read them. They apply to the software named above which includes software licensed by Microsoft Corporation or their affiliates ("Microsoft") to Company. The software also includes documentation and any media on which you received it.

The terms also apply to any

- updates,
- supplements,
- Internet-based services, and
- support services

for this software, unless other terms accompany those items. If so, those terms apply.

By using the software, you accept these terms. If you do not accept them, do not use the software. Instead, contact Company [www.emtrion.com] to determine its return policy. If you comply with these license terms, you have the rights below.

- 1. INSTALLATION AND USE RIGHTS.** You may install and use one copy of the software on each computer on your premises to design, develop, test and demonstrate your programs running validly licensed copies of Microsoft Visual Studio 2005, Team Suite, Visual Studio 2005 Team Edition for Software Architects, Visual Studio 2005 Team Edition for Software Developers, Visual Studio 2005 Team Edition for Software Testers, Visual Studio 2005 Professional Edition, Visual Studio 2005 Tools for the Microsoft Office System or Visual Studio 2005 Standard Edition (including evaluation versions).
- 2. SPEECH RECOGNITION.** If the software includes speech recognition component(s), you understand that speech recognition is an inherently statistical process and that recognition errors are inherent in the process. Neither Company nor Microsoft or their suppliers shall be liable for any damages arising out of errors in the speech recognition process.
- 3. SCOPE OF LICENSE.** The software is licensed, not sold. This agreement only gives you some rights to use the software. Company and Microsoft reserve all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it only in certain ways. For more information, see the software documentation. Except and only to the extent permitted by applicable law despite these limitations, you may not:
 - work around any technical limitations in the software;
 - reverse engineer, decompile or disassemble the software;
 - make more copies of the software than specified in this agreement;
 - publish the software for others to copy;
 - rent, lease or lend the software;
 - transfer the software or this agreement to any third party; or
 - use the software for commercial software hosting services.
- 4. BACK UP MEDIA.** You may make one backup copy of the software. You may use it only to reinstall the software.
- 5. TRADEMARKS.** This agreement does not grant you any rights in connection with any trademarks or service marks of Company, Microsoft, or their suppliers.

6. EXPORT RESTRICTIONS. The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.

7. PRODUCT SUPPORT. Contact Company for support options. Refer to the support number provided with the device. Microsoft does not provide support services for components provided to you in this software.

8. HIGH RISK ACTIVITIES. The software is not fault-tolerant and is not designed, manufactured or intended for any use requiring fail-safe performance in which the failure of the software could lead to death, serious personal injury or severe physical and environmental damage ("High Risk Activities"), such as the operation of aircraft or nuclear facilities. You agree not to use, or license the use of, the software in connection with any High Risk Activities. Company customized this software and is responsible for how it operates.

9. Entire Agreement. This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the software and support services.

10. Applicable Law.

United States. If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

Outside the United States. If you acquired the software in any other country, the laws of that country apply.

11. Legal Effect. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

12. NO WARRANTIES FOR THE SOFTWARE. The software is provided "as is". You bear all risks of using it. Microsoft gives no express warranties, guarantees or conditions. When allowed by your local laws, Company and Microsoft exclude implied warranties of merchantability, fitness for a particular purpose and non-infringement. If your local laws give you any implied warranties, guarantees or conditions, despite this exclusion, your remedies are limited to Company and its suppliers using reasonable efforts to repair or replace the software to the extent permitted by your local laws. Any warranties you receive regarding the software do not originate from, and are not binding on, Microsoft or their affiliates.

13. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. You can recover from Microsoft and their suppliers only direct damages up to U.S. \$5.00. Except for any refund Company may provide, you cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.

It also applies even if

- anything related to the software, services, content (including code) on third party internet sites, or third party programs, and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.
- Company or Microsoft knew or should have known about the possibility of the damages.

The above limitation may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this software is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce logiciel étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le logiciel visé par une licence est offert « tel quel ». Toute utilisation de ce logiciel est à votre seule risque et péril. L'entreprise ou Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection des consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES.

Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Cette indemnisation ne couvre pas les réductions qu'une entreprise vous ait données. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation s'applique aussi à :

- tout ce qui est relié au logiciel, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.
- Ce que l'entreprise ou Microsoft savaient ou auraient du savoir sur les dommages-intérêts.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

15. ADDITIONAL TERMS (if applicable).**End User License Agreement**

Emtrion GmbH Greschbachstrasse 12, 76229 Karlsruhe, Germany (hereinafter referred to as "**Emtrion**") provides to businesspersons which are "Unternehmer" in the sense of Sect. 14 BGB German Civil Code (hereinafter: "the **Customer**") certain Software to be run as embedded software on hardware boards (hereinafter: "the Software") subject to Emtrion's Terms and Conditions:

I. Definitions

1. "**Software**" shall mean the operating software to be embedded into the Hardware, including any documentation hereto.
2. "**Confidential Information**" shall mean any information, know-how and data owned or controlled by Emtrion or by Third Parties license Rights.

II. Limited Use Rights

With respect to Software, Customer is granted a non-exclusive, non-transferable, non-sub licensable, worldwide, license to use or have used the Software for validating or testing. Against an extra fee and an advanced EULA Customer can use Software for projects.

Customer shall not - and shall prevent others from doing so - copy, translate, modify, create derivative works, disassemble, reverse engineer, decompile or otherwise use the Software, except as specifically authorized hereunder in this item, or except provided otherwise by compulsory law. Whenever compulsory law permits the above actions, Recipient shall in any case

- (i) inform Emtrion of these actions in advance, and
- (ii) request support of Emtrion, which Emtrion may offer against reasonable consideration.

All rights granted to Software hereunder shall only apply to the code format to be provided as agreed upon, which is, if not explicitly agreed upon otherwise, solely the binary code to the Software.

Customer is aware that Software may consist of or contains third party software, distributed by Emtrion in altered or unaltered form. The use of such Software may be further restricted by additional terms and conditions (hereinafter also "**Third Parties License Rights**"), with priority to the terms set out herein.

III. ADDITIONAL RESTRICTION ON USE

THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS ARE NOT FAULT TOLERANT AND ARE NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE AS ONLINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

IV. Confidentiality Restrictions

1. **Restrictions.** Without limiting and in addition to the use restrictions under Sect. II and Sect III above, all Confidential Information delivered pursuant to this agreement shall
 - (a) be used by Customer for the Purpose and subject to Section II above only; and
 - (b) be treated by Customer with the same degree of care to avoid unauthorized disclosure to any third party as with respect to Customer's own confidential information of like importance but with no less than the degree of care described in this agreement; and
 - (c) not be distributed, disclosed, or disseminated in any way or form by Customer to anyone except its own employees, who have a reasonable need to know such Confidential Information with respect to the Purpose and who are bound to confidentiality by written agreements not less stringent than under the obligations of this agreement; and
 - (d) be stored in secure areas only that are protected against unauthorized access and are regularly monitored to assure their sufficient security,
 - (e) be treated by Customer in compliance with any additional security guidelines the parties may agree.
2. **Exceptions.** The obligations as per Sect. IV. 1 above shall not apply to any information which the Customer can prove,
 - (a) is at the time of disclosure already in the public domain or becomes available to the public through no breach by the Customer of this agreement, except that Confidential Information shall not be deemed to be in the public domain merely because any part of the Confidential Information is embodied in general disclosures or because individual features, components or combinations thereof are now or become known to the public;
 - (b) is received by the Customer from a third party free to lawfully disclose such information to Customer;
 - (c) was in the Customer's lawful possession prior to receipt from the Discloser as evidenced by written documentation;
 - (d) is independently developed by the Customer without the benefit of any of the Confidential Information as evidenced by written documentation;
 - (e) is approved for release by written agreement of Emtrion;
 - (f) is required to be disclosed to comply with legal mandatory regulations, a judicial or official order or decree, provided that written advance notice of such judicial action was timely given to Emtrion.

V. Prohibition on Development of Competing Products

Customer shall at no time during the term of this agreement and hereafter for a term of five (5) years use Confidential Information for the purpose of developing or distributing products which are competing with Emtrion's Hardware as a stand alone product.

VI. WARRANTIES.

Emtrion makes no warranties, either express or implied, with respect to the object code or source code or with respect to Software or other products or intellectual property obtained from Third Parties. Emtrion expressly disclaims any such warranties, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

